M 1

```
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

D 16

```
**FILE**ID**BASGETRFA






BBBBBBBB      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR      FFFFFFFFFF    AAAAAA
BBBBBBBB      AAAAAA      SSSSSSSS    GGGGGGGG  EEEEEEEEEE  TTTTTTTTTT  RRRRRRRR      FFFFFFFFFF    AAAAAA
BB      BB  AA      AA  SS          GG          EE              TT      RR      RR  FF          AA      AA
BB      BB  AA      AA  SS          GG          EE              TT      RR      RR  FF          AA      AA
BB      BB  AA      AA  SS          GG          EE              TT      RR      RR  FF          AA      AA
BB      BB  AA      AA  SS          GG          EE              TT      RR      RR  FF          AA      AA
BBBBBBBB    AA      AA    SSSSSS    GG          EEEEEEEE        TT      RRRRRRRR    FFFFFFFF    AA      AA
BBBBBBBB    AA      AA    SSSSSS    GG          EEEEEEEE        TT      RRRRRRRR    FFFFFFFF    AA      AA
BB      BB  AAAAAAAAAA          SS  GG  GGGGGG  EE              TT      RR  RR      FF          AAAAAAAAAA
BB      BB  AAAAAAAAAA          SS  GG  GGGGGG  EE              TT      RR  RR      FF          AAAAAAAAAA
BB      BB  AA      AA          SS  GG      GG  EE              TT      RR    RR    FF          AA      AA      ....
BB      BB  AA      AA          SS  GG      GG  EE              TT      RR    RR    FF          AA      AA      ....
BBBBBBBB    AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      RR      RR  FF          AA      AA      ....
BBBBBBBB    AA      AA  SSSSSSSS      GGGGGG    EEEEEEEEEE      TT      RR      RR  FF          AA      AA      ....

LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE BAS$GETRFA (                                  ! Get RFA from RAB
0                 IDENT = '1-004'           ! File: BASGETRFA.B32 Edit: KC1004
0                 ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     Basic Language Support
1 !
1 ! ABSTRACT:
1 !
1 !       BAS$GETRFA will fetch the RFA stored in the RAB.
1 !
1 ! ENVIRONMENT:  Runs at any access mode - AST reentrant
1 !
1 ! AUTHOR: Pamela Levesque, CREATION DATE: 2-Jun-1982
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  PLL 02-Jun-1982
1 ! 1-002 - Make routine global.  PLL 3-Jun-1982
1 ! 1-003 - RFA is passed by ref.  Also, give an error if there is no rfa.
1 !         PLL 4-Jun-1982
1 ! 1-004 - Set up ISB$A_USER_FP so the unwind in the error handler works
1 !         properly.  KC 12-Jun-1984.
1 !--
1
```

```
1 %SBTTL 'Declarations'
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1 !       NONE
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$GETRFA : NOVALUE;                                     ! Get RFA from RAB
1
1 !
1 ! INCLUDE FILES:
1 !
1
1 LIBRARY 'RTLSTARLE';                                  ! System symbols, typically from SYS$LIBRARY:STARLET.L32
1
1 REQUIRE 'RTLIN:RTLPSECT';                             ! Define PSECT declarations macros
1
1 REQUIRE 'RTLML:OTSISB';
1
1 REQUIRE 'RTLML:OTSLUB';
1
1 REQUIRE 'RTLIN:OTSLNK';
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1 !       NONE
1 !
1 ! FIELDS:
1 !
1 !       NONE
1 !
1 ! PSECTS:
1 !
1 DECLARE_PSECTS (BAS);                                 ! Declare PSECTs for BAS$ facility
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
1 !
1
1 EXTERNAL ROUTINE
1       BAS$$STOP_IO : NOVALUE,                         ! Signal fatal I/O error
```

```
;  109      0936  1         BAS$$CB_PUSH : JSB_CB_PUSH NOVALUE,      ! Load register CCB
;  110      0937  1         BAS$$CB_POP : JSB_CB_POP NOVALUE;        ! Done with register CCB
;  111      0938  1
;  112      0939  1 EXTERNAL LITERAL                                 ! Condition value symbols
;  113      0940  1         BAS$K_NO_CURREC : UNSIGNED (8),          ! No current record
;  114      0941  1         BAS$K_ILLILLACC : UNSIGNED (8),          ! Illegal or illogical access
;  115      0942  1         BAS$K_IO_CHANOT : UNSIGNED (8);          ! I/O channel not open
;  116      0943  1
```

```
1 %SBTTL 'BAS$GETRFA - Get RFA from RAB'
1 GLOBAL ROUTINE BAS$GETRFA (                                    ! Get RFA from RAB
1           UNIT,                                                ! logical unit number
1           RFA                                                  ! addr in which to return RFA
1     ) : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine returns the RFA (record file address) of the last record
1 !         accessed for the specified channel.  The 6 byte RFA is stored in the
1 !         location passed as a dtype z descriptor.
1 !
1 ! CALLING SEQUENCE:
1 !
1 !         BAS$GETRFA (UNIT.rlu.v, RFA.wx.r)
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         UNIT.rlu.v        logical unit number
1 !         RFA.wx.r          where to store the RFA
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !         Signals any errors
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !
1 !--
1
2     BEGIN
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     BUILTIN
2         FP;
2
2     LOCAL
2         FMP : REF BLOCK [, BYTE];
2
2     FMP = .FP;
2 !+
2 ! Allocate the LUB/ISB/RAB for this unit if necessary.
2 !-
2     BAS$$CB_PUSH (.UNIT, LUB$K_ILUN_MIN);
2
```

```
;  175           1001  2 !+
;  176           1002  2 ! Load the CCB with the user's FP so that an unwind in the
;  177           1003  2 ! error handler works properly.
;  178           1004  2 !-
;  179           1005  2     CCB [ISB$A_USER_FP] = .FMP [SF$L_SAVE_FP];
;  180           1006  2 !+
;  181           1007  2 ! If the channel is not open, give an error.  There's no RFA for
;  182           1008  2 ! channel 0.
;  183           1009  2 !-
;  184           1010  2     IF (NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
;  185           1011  2
;  186           1012  2 !+
;  187           1013  2 ! No RFA for virtual files.
;  188           1014  2 !-
;  189           1015  2     IF .CCB [LUB$V_VA_USE] THEN BAS$$STOP_IO (BAS$K_ILLILLACC);
;  190           1016  2
;  191           1017  2 !+
;  192           1018  2 ! Return the RFA.
;  193           1019  2 !-
;  194           1020  2     IF .CCB [RAB$L_RFA0] NEQ 0
;  195           1021  2     THEN
;  196           1022  2         CH$MOVE (6, CCB [RAB$W_RFA], .RFA)
;  197           1023  2     ELSE
;  198           1024  3         BEGIN
;  199           1025  3         BAS$$STOP_IO (BAS$K_NO_CURREC)
;  200           1026  2         END;
;  201           1027  2 !+
;  202           1028  2 ! Pop the CCB off the I/O system.
;  203           1029  2 !-
;  204           1030  2     BAS$$CB_POP ();
;  205           1031  2
;  206           1032  1     END;                                                   ! End of routine BAS$GETRFA

                                                                        .TITLE  BAS$GETRFA
                                                                        .IDENT  \1-004\

                                                                        .EXTRN  BAS$$STOP_IO, BAS$$CB_PUSH
                                                                        .EXTRN  BAS$$CB_POP, BAS$K_NO_CURREC
                                                                        .EXTRN  BAS$K_ILLILLACC
                                                                        .EXTRN  BAS$K_IO_CHANOT

                                                                        .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                                 087C 00000             .ENTRY  BAS$GETRFA, Save R2,R3,R4,R5,R6,R11            ; 0945
                         56 00000000G    00   9E 00002                  MOVAB   BAS$$STOP_IO, R6                                ;
                         53              5D   D0 00009                  MOVL    FP, FMP                                         ; 0995
                         50              08   CE 0000C                  MNEGL   #8, R0                                          ; 0999
                         52        04    AC   D0 0000F                  MOVL    UNIT, R2                                        ;
                            00000000G    00   16 00013                  JSB     BAS$$CB_PUSH                                    ;
                  FF4C   CB        0C    A3   D0 00019                  MOVL    12(FMP), -180(CCB)                              ; 1005
                         07        FC    AB   E8 0001F                  BLBS    -4(CCB), 1$                                     ; 1010
                         7E       00G    8F   9A 00023                  MOVZBL  #BAS$K_IO_CHANOT, -(SP)                         ;
                         66              01   FB 00027                  CALLS   #1, BAS$$STOP_IO                                ;
                         07        FF    AB   E9 0002A 1$:              BLBC    -1(CCB), 2$                                     ; 1015
                         7E       00G    8F   9A 0002E                  MOVZBL  #BAS$K_ILLILLACC, -(SP)                         ;
                         66              01   FB 00032                  CALLS   #1, BAS$$STOP_IO                                ;
```

```
                                10    AB  D5 00035 2$:     TSTL    16(CCB)                              ; 1020
                                      08  13 00038         BEQL    3$                                   ;
          08    BC      10    AB      06  28 0003A         MOVC3   #6, 16(CCB), @RFA                    ; 1022
                                      07  11 00040         BRB     4$                                   ;
                              7E  00G 8F  9A 00042 3$:     MOVZBL  #BAS$K_NO_CURREC, -(SP)              ; 1025
                              66      01  FB 00046         CALLS   #1, BAS$$STOP_IO                     ;
                           00000000G  00  16 00049 4$:     JSB     BAS$$CB_POP                          ; 1030
                                          04 0004F         RET                                          ; 1032

; Routine Size:  80 bytes,    Routine Base:  _BAS$CODE + 0000

;  207          1033  1 !<BLF/PAGE>
;  209          1034  1 END                                                ! End of module BAS$GETRFA
;  210          1035  1
;  211          1036  0 ELUDOM




;                                       PSECT SUMMARY
;
;       Name                      Bytes                    Attributes
;
;  _BAS$CODE                           80  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                               Library Statistics
;
;                                            --------- Symbols --------          Pages         Processing
;       File                                 Total     Loaded   Percent          Mapped        Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1          9776         3         0             581           00:01.0




;                                        COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BASGETRFA/OBJ=OBJ$:BASGETRFA MSRC$:BASGETRFA/UPDATE=(ENH$:BASGETRFA
;       )

; Size:           80 code + 0 data bytes
; Run Time:          00:08.5
; Elapsed Time:      00:20.3
; Lines/CPU Min:      7338
; Lexemes/CPU-Min: 43912
; Memory Used:  115 pages
; Compilation Complete
```

0023 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASEXITHA
LIS

BASFETCHA
LIS

BASFETCHD
LIS

BASFIND
LIS

BASFORINT
LIS

BASFORMAT
LIS

BASFREE
LIS

BASFSP
LIS

BASGET
LIS

BASGETRFA
LIS

BASHANDLE
LIS